Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	(cache adj purge adj instruction) and (cache adj line) and (plurality near3 processors)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:13

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	346	711/135.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/19 23:58
L2	171	1 and invalidat$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:10
L3	36	2 and (updat$4 with (flush$4 or purg$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:10
L4	21781	(purg$4 or flush$3 or dump$3 or evict$3) same (modif$7 or updat$3)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 00:32
L6	778	4 same (cache adj line)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 00:32
L7	80336	multiprocessor or (plurality near4 processor) or (multi adj processor)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 00:33
L8	18980	shared adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 00:33
L9	218	6 and 7 and 8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 00:34
L11	24	(send or forward or sent or forwarded) near3 (updat$3 or modified) near3 (cache adj line) near10 (processor or node)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 00:36

L12	7	9 and 11	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:09
L13	3355	(711/141 or 711/143 or 711/156 or 711/163).ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:10
L14	1149	13 and invalidat$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:10
L15	102	14 and (updat$4 with (flush$4 or purg$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:10
L16	71	15 and (cache adj line)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:11
L17	29	7 and 8 and 16	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2005/12/20 01:11

Terms used
invalidate and **update** or **flush** or **purge** and **cache** **line** and **multiprocessors** and **shared** **memory**

Found **15,586** of **167,655**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Parallel and object oriented programming: Integrating applications with cache and memory management on a shared-memory multiprocessor

Harjinder S. Sandhu

November 1992 **Proceedings of the 1992 conference of the Centre for Advanced Studies on Collaborative research - Volume 1**

Publisher: IBM Press

Full text available: pdf(823.01 KB) Additional Information: full citation, abstract, references, citings

In shared memory multiprocessors with NonUniform Memory Access (NUMA) characteristics, effective cacheing and memory locality are essential to performance. In this paper, we argue for a new approach for cache and NUMA memory management based upon the integration of application-sharing characteristics with system runtime management of shared data. An application's shared data is subdivided into *shared regions* of memory, and the application defines explicitly the operations on those regions ...

2 The shared regions approach to software cache coherence on multiprocessors

Harjinder S. Sandhu, Benjamin Gamsa, Songnian Zhou

July 1993 **ACM SIGPLAN Notices , Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming PPOPP '93**, Volume 28 Issue 7

Publisher: ACM Press

Full text available: pdf(1.12 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The effective management of caches is critical to the performance of applications on shared-memory multiprocessors. In this paper, we discuss a technique for software cache coherence tht is based upon the integration of a program-level abstraction for shared data with software cache management . The program-level abstraction, called Shared Regions, explicitly relates synchronization objects with the data they protect. Cache coherence algorithms are presented which use the i ...

3 Hierarchical cache/bus architecture for shared memory multiprocessors

A. W. Wilson

June 1987 **Proceedings of the 14th annual international symposium on Computer architecture**

Publisher: ACM Press

Full text available: pdf(978.39 KB) Additional Information: full citation, abstract, references, citings, index terms

A new, large scale multiprocessor architecture is presented in this paper. The architecture consists of hierarchies of shared buses and caches. Extended versions of shared bus multicache coherency protocols are used to maintain coherency among all caches in the system. After explaining the basic operation of the strict hierarchical approach, a clustered system is introduced which distributes the memory among groups of processors. Results of simulations are presented which demonstrate that t ...

4 Hardware fault containment in scalable shared-memory multiprocessors

Dan Teodosiu, Joel Baxter, Kinshuk Govil, John Chapin, Mendel Rosenblum, Mark Horowitz

May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture ISCA '97**, Volume 25 Issue 2

Publisher: ACM Press

Full text available: pdf(2.05 MB) Additional Information: full citation, abstract, references, citings, index terms

Current shared-memory multiprocessors are inherently vulnerable to faults: any significant hardware or system software fault causes the entire system to fail. Unless provisions are made to limit the impact of faults, users will perceive a decrease in reliability when they entrust their applications to larger machines. This paper shows that fault containment techniques can be effectively applied to scalable shared-memory multiprocessors to reduce the reliability problems created by increased mach ...

5 Cache coherence for large scale shared memory multiprocessors

M. Thapar, B. Delagi

May 1990 **Proceedings of the second annual ACM symposium on Parallel algorithms and architectures**

Publisher: ACM Press

Full text available: pdf(645.67 KB) Additional Information: full citation, references, citings, index terms

6 Performance evaluation of a commercial cache-coherent shared memory multiprocessor

Rajeev Jog, Philip L. Vitale, James R. Callister

April 1990 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems SIGMETRICS '90**, Volume 18 Issue 1

Publisher: ACM Press

Full text available: pdf(948.46 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes an approximate Mean Value Analysis (MVA) model developed to project the performance of a small-scale shared-memory commercial symmetric multiprocessor system. The system, based on Hewlett Packard Precision Architecture processors, supports multiple active user processes and multiple execution threads within the operating system. Using detailed timing for hardware delays, a customized approximate closed queueing model is developed for the multiprocessor system ...

7 Architectural support for scalable speculative parallelization in shared-memory multiprocessors

Marcelo Cintra, José F. Martínez, Josep Torrellas

May 2000 **ACM SIGARCH Computer Architecture News , Proceedings of the 27th annual international symposium on Computer architecture ISCA '00**, Volume 28 Issue 2

Publisher: ACM Press

Full text available: pdf(253.29 KB) Additional Information: full citation, abstract, references, citings, index terms

Speculative parallelization aggressively executes in parallel codes that cannot be fully parallelized by the compiler. Past proposals of hardware schemes have mostly focused on single-chip multiprocessors (CMPs), whose effectiveness is necessarily limited by their small size. Very few schemes have attempted this technique in the context of scalable shared-memory systems. In this paper, we present and evaluate a new hardware scheme for scalable speculative parallelization. This de ...

8 Cache coherence for large scale shared memory multiprocessors

Manu Thapar, Bruce Delagi

March 1991 **ACM SIGARCH Computer Architecture News**, Volume 19 Issue 1

Publisher: ACM Press

Full text available: pdf(534.44 KB) Additional Information: full citation, index terms

9 Architectural primitives for a scalable shared memory multiprocessor

Joonwon Lee, Umakishore Ramachandran

June 1991 **Proceedings of the third annual ACM symposium on Parallel algorithms and architectures**

Publisher: ACM Press

Full text available: pdf(1.27 MB) Additional Information: full citation, references, citings, index terms

10 CRL: high-performance all-software distributed shared memory

K. L. Johnson, M. F. Kaashoek, D. A. Wallach

December 1995 **ACM SIGOPS Operating Systems Review , Proceedings of the fifteenth ACM symposium on Operating systems principles SOSP '95**, Volume 29 Issue 5

Publisher: ACM Press

Full text available: pdf(2.02 MB) Additional Information: full citation, references, citings, index terms

11 The directory-based cache coherence protocol for the DASH multiprocessor

Daniel Lenoski, James Laudon, Kourosh Gharachorloo, Anoop Gupta, John Hennessy

May 1990 **ACM SIGARCH Computer Architecture News , Proceedings of the 17th annual international symposium on Computer Architecture ISCA '90**, Volume 18 Issue 3a

Publisher: ACM Press

Full text available: pdf(1.74 MB) Additional Information: full citation, abstract, references, citings, index terms

DASH is a scalable shared-memory multiprocessor currently being developed at Stanford's Computer Systems Laboratory. The architecture consists of powerful processing nodes, each with a portion of the shared-memory, connected to a scalable interconnection network. A key feature of DASH is its distributed directory-based cache coherence protocol. Unlike traditional snoopy coherence protocols, the DASH protocol does not rely on broadcast; instead it uses point-to-point messages sent between th ...

12 Multiprocessor cache synchronization: issues, innovations, evolution

P. Bitar, A. M. Despain

June 1986 **ACM SIGARCH Computer Architecture News , Proceedings of the 13th annual international symposium on Computer architecture ISCA '86**, Volume

14 Issue 2
Publisher: IEEE Computer Society Press, ACM Press

Full text available: pdf(981.69 KB) Additional Information: full citation, abstract, references, citings, index terms

Many options are possible in a cache synchronization (or consistency) scheme for a broadcast system. We clarify basic concepts, analyze the handling of shared data, and then describe a protocol that we are currently exploring. Finally, we analyze the evolution of options that have been proposed under write-in (or write-back) policy. We show how our protocol extends this evolution with new methods for efficient busy-wait locking, waiting, and unlocking. The ...

13 ReViVe: cost-effective architectural support for rollback recovery in shared-memory multiprocessors

Milos Prvulovic, Zheng Zhang, Josep Torrellas

May 2002 **ACM SIGARCH Computer Architecture News , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02 , Proceedings of the 29th annual international symposium on Computer architecture ISCA '02**, Volume 30 Issue 2

Publisher: IEEE Computer Society, ACM Press

Full text available: pdf(1.38 MB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

This paper presents ReVive, a novel general-purpose rollback recovery mechanism for shared-memory multiprocessors. ReVive carefully balances the conflicting requirements of availability, performance, and hardware cost. ReVive performs checkpointing, logging, and distributed parity protection, all memory-based. It enables recovery from a wide class of errors, including the permanent loss of an entire node. To maintain high performance, ReVive includes specialized hardware that performs frequent o ...

Keywords: fault tolerance, shared-memory multiprocessors, rollback recovery, recovery, BER, logging, parity, checkpointing, availability

14 Compiler and hardware support for cache coherence in large-scale multiprocessors: design considerations and performance study

Lynn Choi, Pen-Chung Yew

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper, we study a hardware-supported, compiler directed (HSCD) cache coherence scheme, which can be implemented on a large-scale multiprocessor using off-the-shelf microprocessors, such as the Cray T3D. It can be adapted to various cache organizations, including multi-word cache lines and byte-addressable architectures. Several system related issues, including critical sections, inter-thread communication, and task migration have also been addressed. The cost of the required hardware sup ...

15 Shared memory computing on clusters with symmetric multiprocessors and system area networks

Leonidas Kontothanassis, Robert Stets, Galen Hunt, Umit Rencuzogullari, Gautam Altekar, Sandhya Dwarkadas, Michael L. Scott

August 2005 **ACM Transactions on Computer Systems (TOCS)**, Volume 23 Issue 3

Publisher: ACM Press

Full text available: pdf(918.28 KB) Additional Information: full citation, abstract, references, index terms

Cashmere is a software distributed shared memory (S-DSM) system designed for clusters of server-class machines. It is distinguished from most other S-DSM projects by (1) the effective use of fast user-level messaging, as provided by modern system-area networks, and (2) a "two-level" protocol structure that exploits hardware coherence within multiprocessor nodes. Fast user-level messages change the tradeoffs in coherence protocol design; they allow Cashmere to employ a relatively simp ...

Keywords: Distributed shared memory, relaxed consistency, software coherence

16 Multigrain shared memory

Donald Yeung, John Kubiatowicz, Anant Agarwal

May 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 2

Publisher: ACM Press

Full text available: pdf(369.18 KB) Additional Information: full citation, abstract, references, index terms, review

Parallel workstations, each comprising tens of processors based on shared memory, promise cost-effective scalable multiprocessing. This article explores the coupling of such small- to medium-scale shared-memory multiprocessors through software over a local area network to synthesize larger shared-memory systems. We call these systems Distributed Shared-memory MultiProcessors (DSMPs). This article introduces the design of a shared-memory system that uses multiple granularities of sharing, ca ...

Keywords: distributed memory, symmetric multiprocessors, system of systems

17 MGS: a multigrain shared memory system

Donald Yeung, John Kubiatowicz, Anant Agarwal

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.37 MB) Additional Information: full citation, abstract, references, citings, index terms

Parallel workstations, each comprising 10-100 processors, promise cost-effective general-purpose multiprocessing. This paper explores the coupling of such small- to medium-scale shared memory multiprocessors through software over a local area network to synthesize larger shared memory systems. We call these systems Distributed Scalable Shared-memory Multiprocessors (DSSMPs).This paper introduces the design of a shared memory system that uses multiple granularities of sharing, and presents an imp ...

18 COMA: an opportunity for building fault-tolerant scalable shared memory multiprocessors

Christine Morin, Alain Gefflaut, Michel Banâtre, Anne-Marie Kermarrec

May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.30 MB) Additional Information: full citation, abstract, references, citings, index terms

Due to the increasing number of their components, Scalable Shared Memory Multiprocessors (SSMMs) have a very high probability of experiencing failures. Tolerating node failures therefore becomes very important for these architectures particularly if they

must be used for long-running computations. In this paper, we show that the class of Cache Only Memory Architectures (COMA) are good candidates for building fault-tolerant SSMMs. A backward error recovery strategy can be implemented without sign ...

Keywords: Scalable Shared

19 Performance of database workloads on shared-memory systems with out-of-order processors

Parthasarathy Ranganathan, Kourosh Gharachorloo, Sarita V. Adve, Luiz André Barroso

October 1998 **ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the eighth international conference on Architectural support for programming languages and operating systems ASPLOS-VIII**, Volume 33 , 32 Issue 11 , 5

Publisher: ACM Press

Full text available: pdf(1.62 MB) Additional Information: full citation, abstract, references, citings, index terms

Database applications such as online transaction processing (OLTP) and decision support systems (DSS) constitute the largest and fastest-growing segment of the market for multiprocessor servers. However, most current system designs have been optimized to perform well on scientific and engineering workloads. Given the radically different behavior of database workloads (especially OLTP), it is important to re-evaluate key system design decisions in the context of this important class of applicatio ...

20 Recovery protocols for shared memory database systems

Lory D. Molesky, Krithi Ramamritham

May 1995 **ACM SIGMOD Record , Proceedings of the 1995 ACM SIGMOD international conference on Management of data SIGMOD '95**, Volume 24 Issue 2

Publisher: ACM Press

Full text available: pdf(1.65 MB) Additional Information: full citation, abstract, references, index terms

Significant performance advantages can be gained by implementing a database system on a cache-coherent shared memory multiprocessor. However, problems arise when failures occur. A single node (where a *node* refers to a processor/memory pair) crash may require a reboot of the entire shared memory system. Fortunately, shared memory multiprocessors that isolate individual node failures are currently being developed. Even with these, because of the side effects of the cache coherency protocol, ...

Results for "((invalidate*<and>(update*<or>flush*<or>purge*)<and>(cache<near/1>line..."

Your search matched **0** documents.

A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistan search.

OPTION 1
Enter keywords or phrases, select fields, and select operators

Help

[ ] in All Fields

AND [ ] in All Fields

AND [ ] in All Fields

» Note: If you use all three search boxes, the entries in the first two boxes take precedence over the entry in the third box.

OPTION 2
Enter keywords, phrases, or a Boolean expression

Help

```
invalidate*<and>
(update*<or>flush*<or>purge*)<and>
(cache<near/1>line)
<and>multiprocessors<and>
(shared<near/1>memory)
```

» Note: You may use the search operators <and> or <or> without the start and end brackets <>.

» Learn more about Field Codes, Search Examples, and Search Operators

» Publications

Select publications

- IEEE Periodicals
- IEE Periodicals
- IEEE Conference
- IEE Conference P
- IEEE Standards

» Other Resources (Availat

- IEEE Books

» Select date range

Search latest content u

From year All

to Present

» Display Format

Citation Citatio

» Organize results

Maximum 100

Display 25 resi

Sort by Relevance

In Descending

Indexed by Inspec